

Starting with AWX

1- Creating a GitLab SCM Project

1.1- Gitlab Credentials

To use a Git repository as the source for Ansible Playbooks in Awx, you'll need to add your Gitlab credentials. If you are using a privately hosted Gitlab installation, you will first need to generate an SSH keypair. If you are using the Gitlab hosted service, please refer to the Ansible documentation.

```
cd ~/.ssh
ssh-keygen -N '' -f awx_ssh_key
```

Now add your private key to the AWX Source Control Credentials:

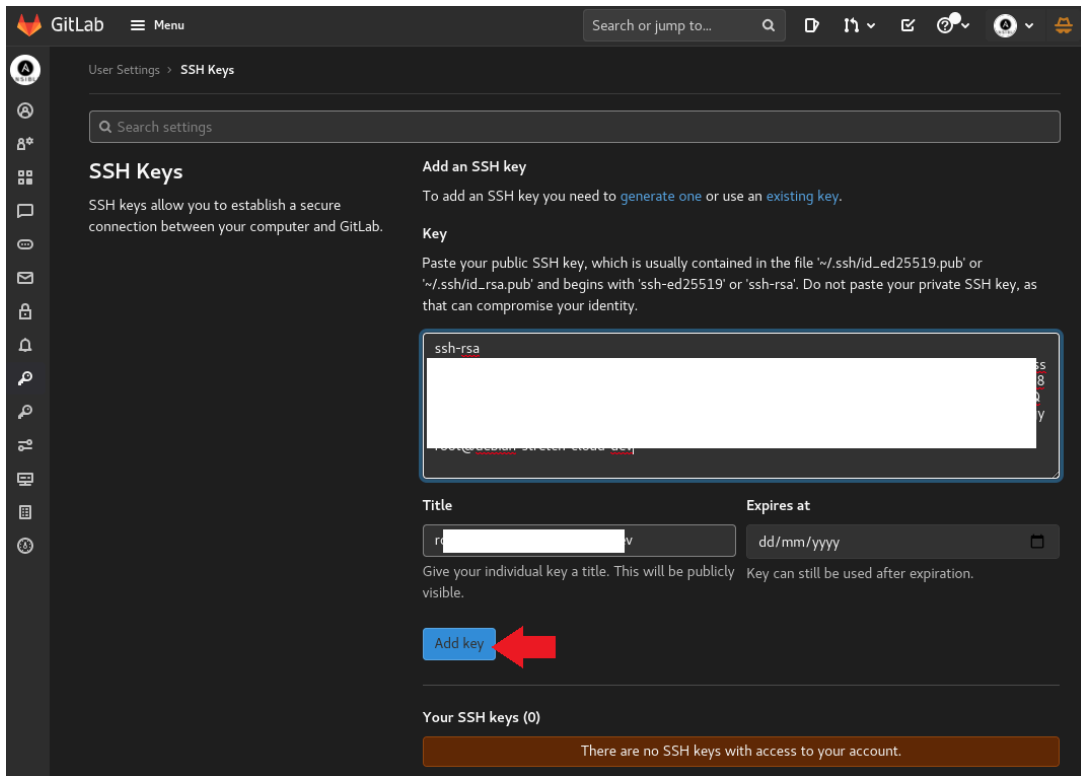
```
cat awx_ssh_key
```

The screenshot shows the 'Create New Credential' form in the AWX web interface. The sidebar on the left contains navigation links for Activity Stream, Workflow Approvals, Resources (Templates, Credentials, Projects, Inventories, Hosts), Access (Organizations, Users, Teams), and Administration (Credential Types, Notifications, Management Jobs, Instance Groups, Applications). The main form area is titled 'Create New Credential' and contains the following fields:

- Name:** Gitlab
- Description:** Gitlab User for Ansible Deployments
- Organization:** (searchable)
- Credential Type:** Source Control
- Type Details:**
 - Username:** (masked)
 - Password:** (masked)
- SCM Private Key:** (file upload area with a redacted key)
- Private Key Passphrase:** (masked)

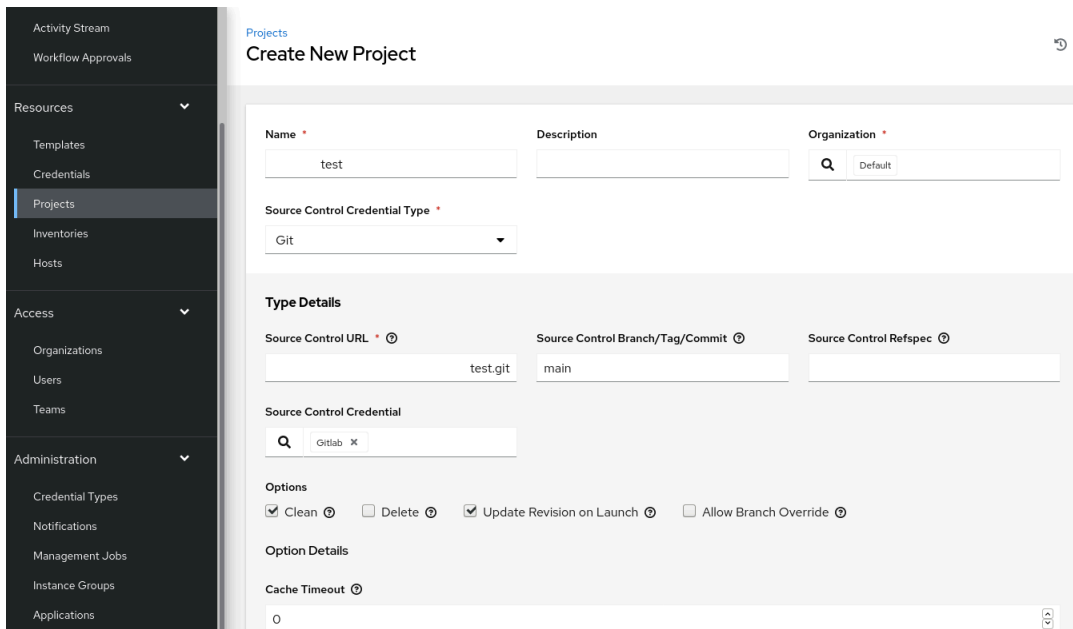
And the public key has to be added to your Gitlab user account:

```
cat awx_ssh_key.pub
```

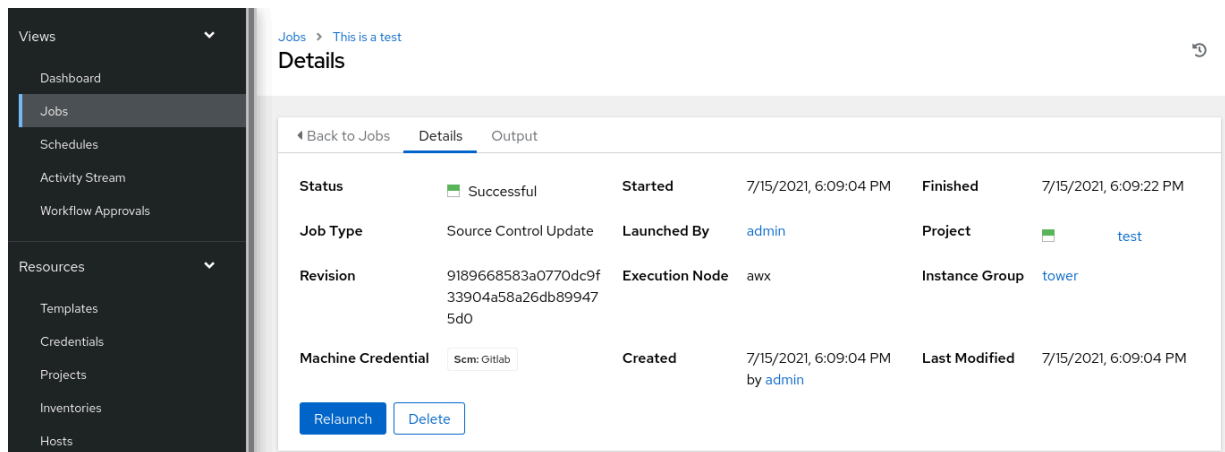


1.2- Creating the Project

The credentials were then used to create a new project. The Source Control URL was set to a newly added test repository on Gitlab.



Save the Project and hit Sync - check under Jobs to see if the run was successful:

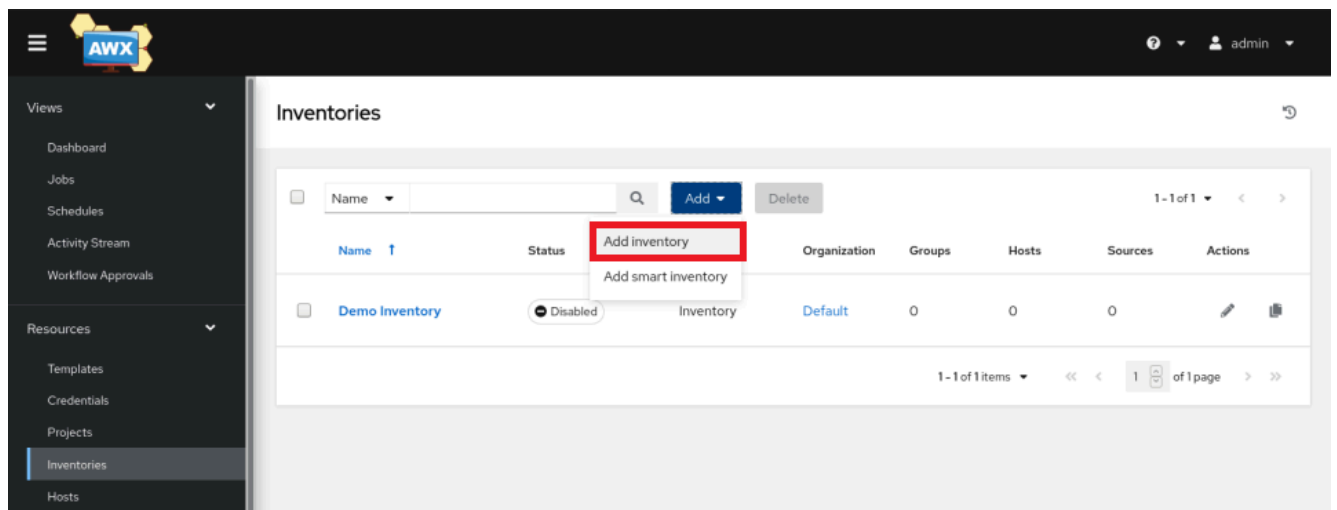


Now that we know that Ansible Tower has access to our Gitlab repo, we can start by building our Ansible Playbook.

We'll start by a simple debug of a Hello World message for the playbook. This will simply try to connect to our server, write a Hello World string into the terminal and return a successful result if everything worked.

1.3- Adding Hosts

To run our playbook we now have to add a host to Ansible Tower. To add a new host to our AWX frontend we need to create an inventory in the **Resources/Inventories** menu:



After saving the inventory, we can add a host from the Host tab:

Inventories > Main inventory > Hosts > MikePlayground

Edit details

Name * Description

Variables **YAML** JSON

```
1 ---
2 ansible_host: 11[REDACTED]4
3 ansible_port: 4[REDACTED]
```

Save Cancel

Now we need to switch to the **Resources/Credentials** menu and add our server login:

AWX

Dashboard
Jobs
Schedules
Activity Stream
Workflow Approvals

Resources

- Templates
- Credentials**
- Projects
- Inventories
- Hosts

Access

Create New Credential

Name * Description Organization

Credential Type *

Machine

Type Details

Username Password ☐ Prompt on launch

root [REDACTED]

SSH Private Key

2- Execute the Playbook

Now that we have all the necessary information for Ansible Tower to work, let's create a Job Template that incorporates everything and test it out.

Details

[Back to Templates](#) **Details** [Access](#) [Notifications](#) [Schedules](#) [Completed Jobs](#) [Survey](#)

Name	Hello World	Job Type	run	Organization	Default
Inventory	Main inventory	Project	test	Playbook	helloworld.yml
Forks	0	Verbosity	0 (Normal)	Timeout	0
Show Changes	Off	Job Slicing	1	Options	Use Fact Storage
Created	7/15/2021, 7:15:47 PM by admin	Last Modified	7/15/2021, 8:28:36 PM by admin		

Credentials SSH: mikes_playground

Variables **YAML** [JSON](#)

1 ---

Edit

Launch

Delete

The job template will be executed once you click "Launch."

Output

[Back to Jobs](#) [Details](#) **Output**

Hello World

Plays 1 Tasks 2 Elapsed 00:00:19

```
+
0 SSH password:
1
2 PLAY [all] ***** 20:29:02
3
4 TASK [Gathering Facts] ***** 20:29:02
5 ok:
6
7 TASK [Hello World!] ***** 20:29:13
8 changed:
9
10 PLAY RECAP ***** 20:29:20
11 Playground : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescue
12 d=0 ignored=0
```